Today’s Agenda

- Mathematical tools in digital image processing
- Intensity Transformation

Distance Measures

For pixels p, q, and z, with coordinates (x,y), (s,t) and (v,w), D is a distance function or metric if

$$(a)\ D(p,q) \geq 0 \quad D(p,q) = 0\ iff\ \ p = q$$

$$(b)\ D(p,q) = D(q,p), \quad and$$

$$(c)\ D(p,z) \leq D(p,q) + D(q,z)$$

Distance Measures

Euclidean distance $$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

City-block (D4) distance $$D_4(p,q) = |x-s| + |y-t|$$

Chessboard (D8) distance (Chebyshev distance)

$$D_8(p,q) = \max(|x-s|, |y-t|)$$

Distance: Sample Problem

D4 distance

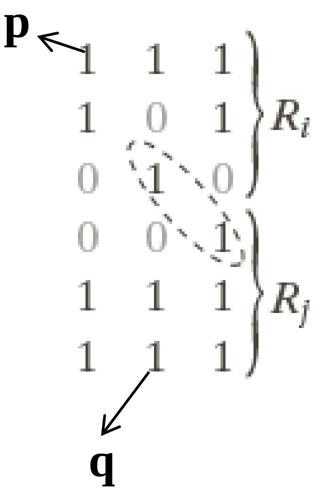
6

D8 distance

5

Euclidean distance

$$\sqrt{\mathbf{1 + 5^2}}$$



Distance vs length of a path?

Mathematic Tools

Array/Matrix operations

Linear/nonlinear operations

Linearity: $H\left[a_i f_i(x, y) + a_j f_j(x, y)\right] = a_i H\left[f_i(x, y)\right] + a_j H\left[f_j(x, y)\right]$

Arithmetic Operations – single pixel operations

- Image averaging, image subtraction, image multiplication

Set and logic operations

Spatial operations

- Single pixel operations and neighborhood operations

Image transformation

Probabilistic methods

Mathematic Tools

Array versus Matrix operations

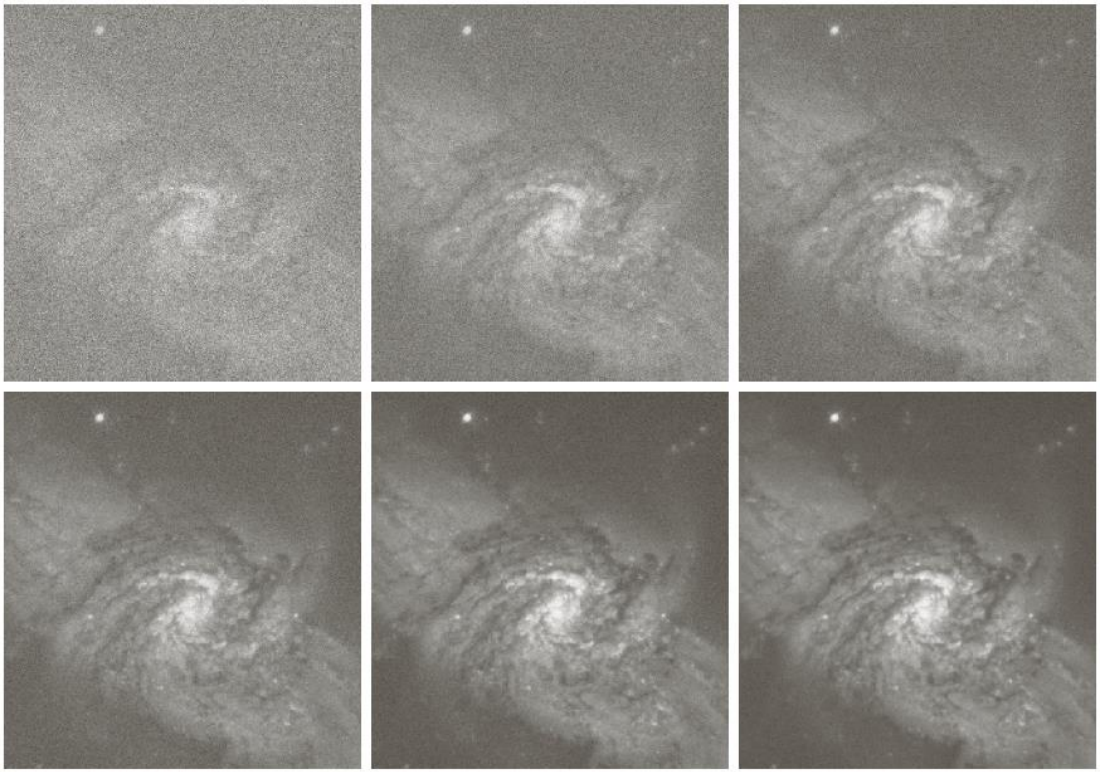
Array Multiplications

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Matrix Multiplications

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Image Averaging – Noise Reduction



a b c
d e f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

$$g(x, y) = f(x, y) + \eta(x, y)$$

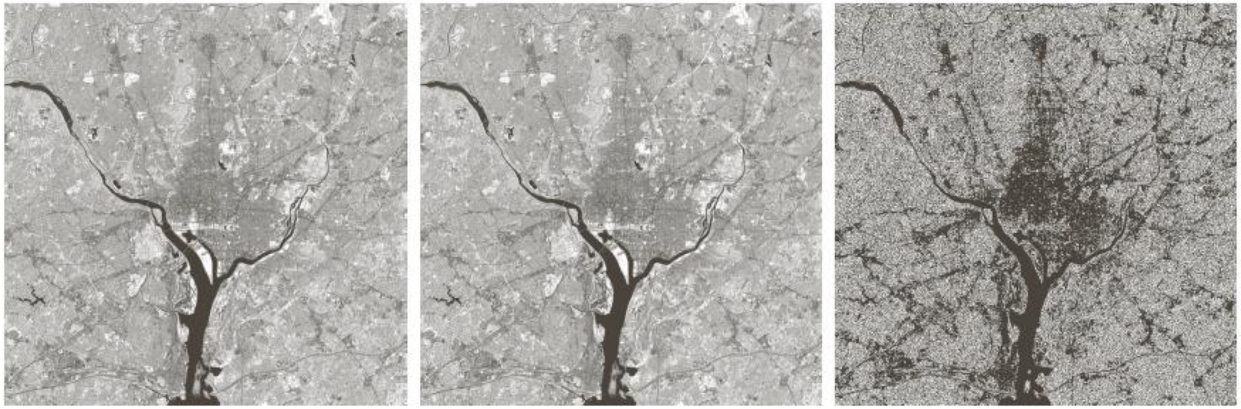
$$\overline{g}(x, y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x, y)$$

$$E\{\overline{g}(x, y)\} = f(x, y)$$

$$\sigma^2_{\overline{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$$

Assumption: the noise is uncorrelated in image and has zero mean

Image Subtraction – Enhance Difference



a b c

FIGURE 2.27 (a) Infrared image of the Washington, D.C. area. (b) Image obtained by setting to zero the least significant bit of every pixel in (a). (c) Difference of the two images, scaled to the range [0, 255] for clarity.

Image Subtraction

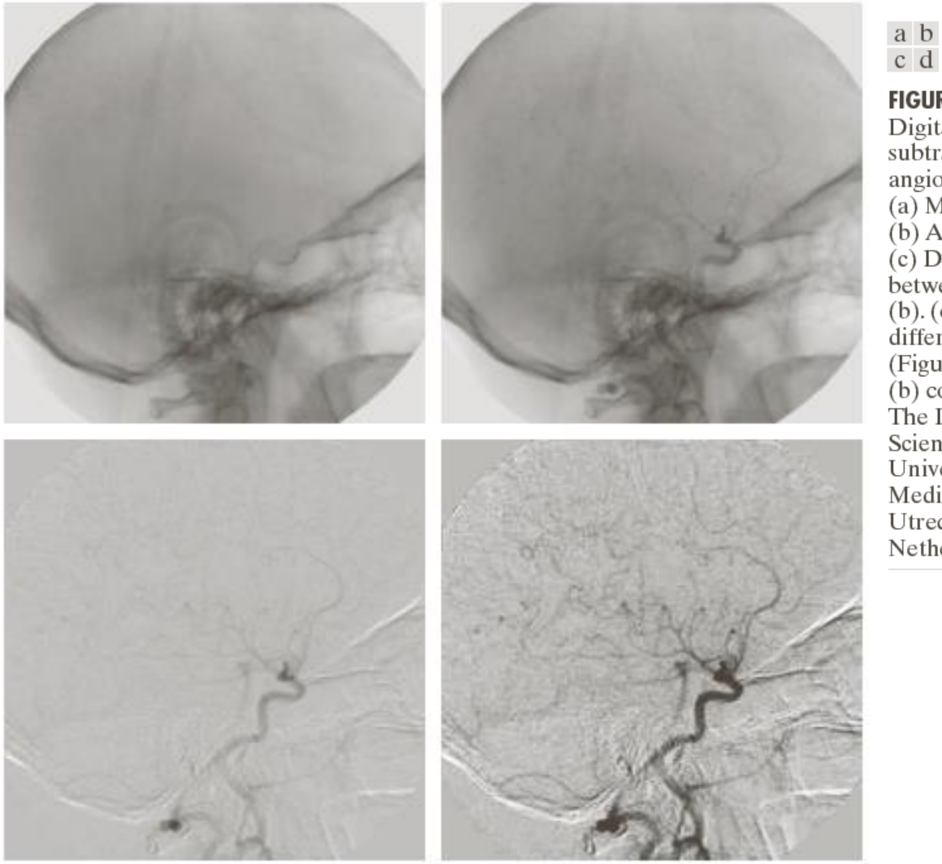


FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

The images used in averaging & subtraction must be registered!

Image Multiplication (Division)

$$g(x,y)=f(x,y)h(x,y)$$

FIGURE 2.30 (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

$$g(x,y)=f(x,y)/h(x,y)$$

FIGURE 2.29 Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Notes on Arithmetic Operations

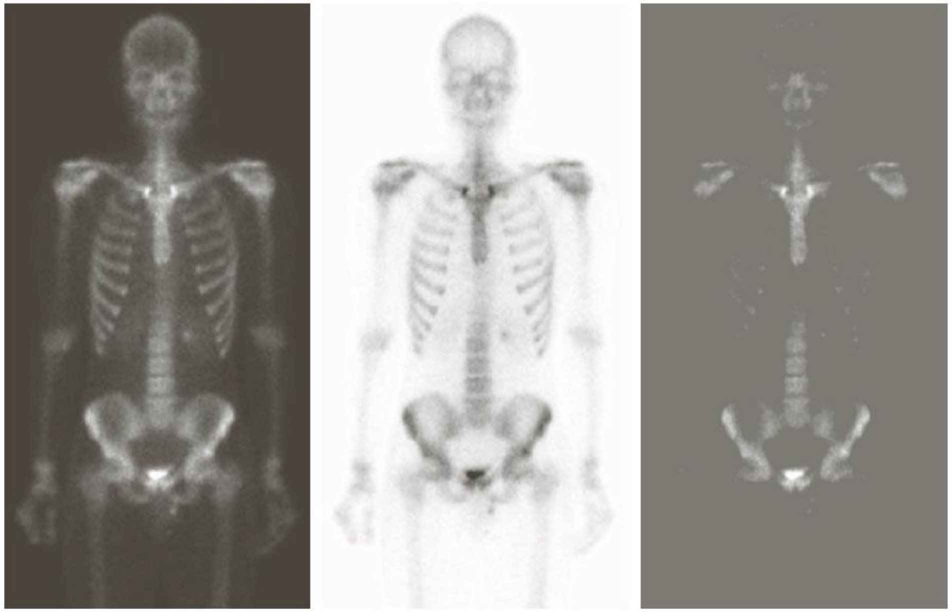
The images used in averaging & subtraction must be registered!

Output images should be normalized to the range of [0,255]

$$f_m = f - \min(f)$$

$$f_s = K[f_m / \max(f_m)]$$

Set Operations Based on Intensities



a b c

FIGURE 2.32 Set operations involving gray-scale images. (a) Original image. (b) Image negative obtained using set complementation. (c) The union of (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

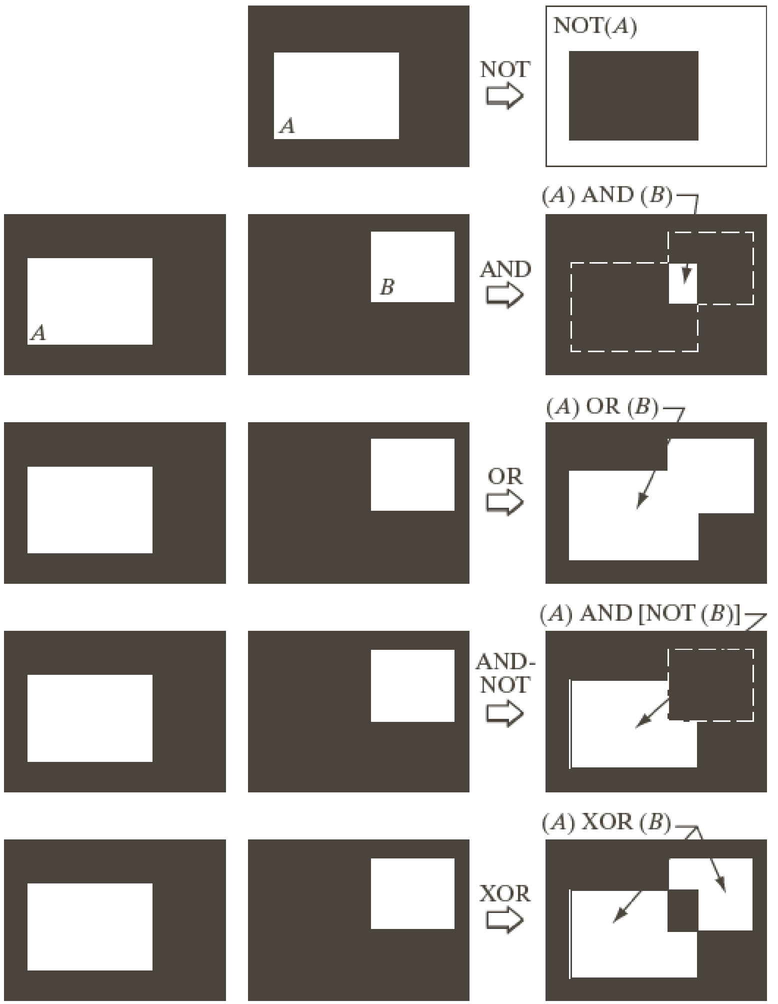
Complement – negative image

$$\boldsymbol{A^c = \{(x, y, K - z) | (x, y, z) \in A\}}$$

Thresholding

$$\boldsymbol{A \cup B = \left\{\left(x, y, \max(z_a, z_b)\right) \middle| (x, y, z_a) \in A, (x, y, z_b) \in B\right\}}$$

Logic Operations for Binary Image



Foreground/background

- **Binary image: 0/1**
- **Fuzzy set: [0,1]**

Logic operations will be used a lot in morphological image processing

FIGURE 2.33 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Spatial Operations

Perform directly on the pixels of the given image

- Intensity transformation – change the intensity
  - Single pixel operations $s=T(z)$
  - Neighborhood operations
- Geometric spatial transformations – change the coordinates

Single pixel operations

- Determined by
  - Transformation function T
  - Input intensity value
- Not depend on other pixels and position

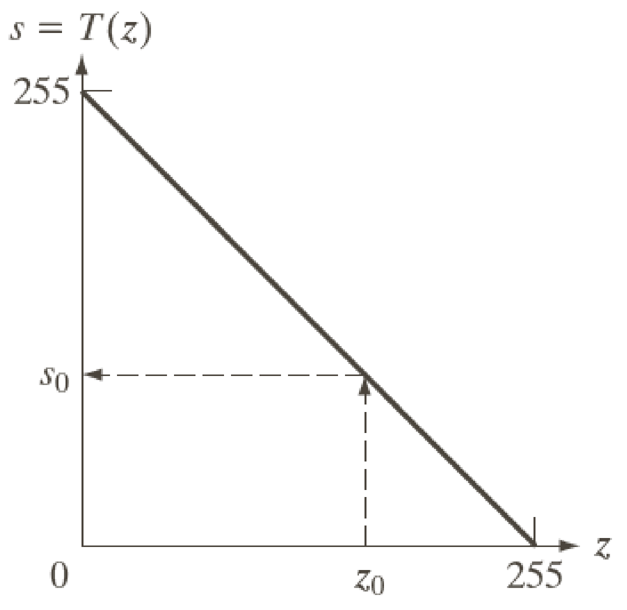


FIGURE 2.34 Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0.

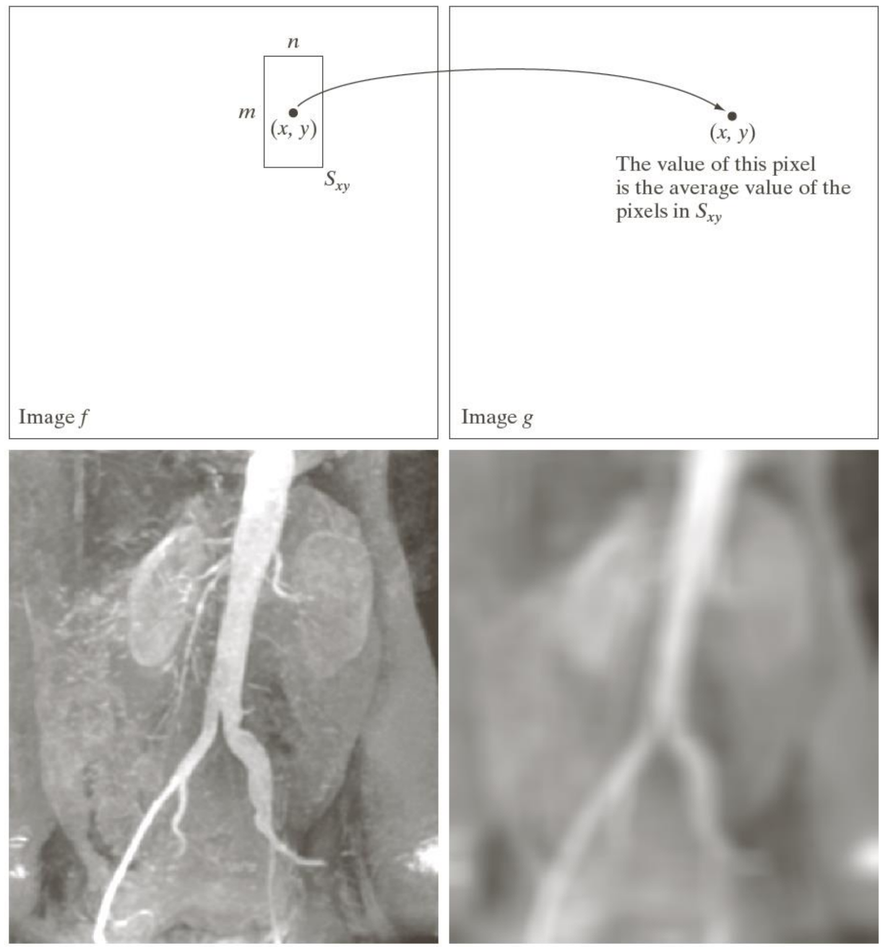
Neighborhood Operations

Image smoothing

$$g(x,y) = \frac{1}{mn} \sum_{(r,c)\in S_{xy}} f(r,c)$$

Other examples:

- Interpolation
- Image filtering



a b
c d

FIGURE 2.35 Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) The aortic angiogram discussed in Section 1.3.2. (d) The result of using Eq. (2.6-21) with $m = n = 41$. The images are of size 790×686 pixels.

Geometric Spatial Transformations – Rubber Sheet Transformation

$$(x, y) = T\{(v, w)\}$$

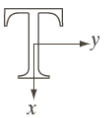
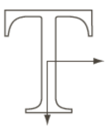
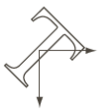
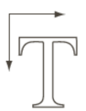
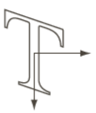
Affine transform:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$

Inverse mapping

$$\begin{bmatrix} v \\ w \\ 1 \end{bmatrix} = \mathbf{T}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

TABLE 2.2
Affine transformations based on Eq. (2.6.–23).

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Geometric Spatial Transformations

Note: a neighborhood operation, i.e., interpolation, is required following geometric transformation

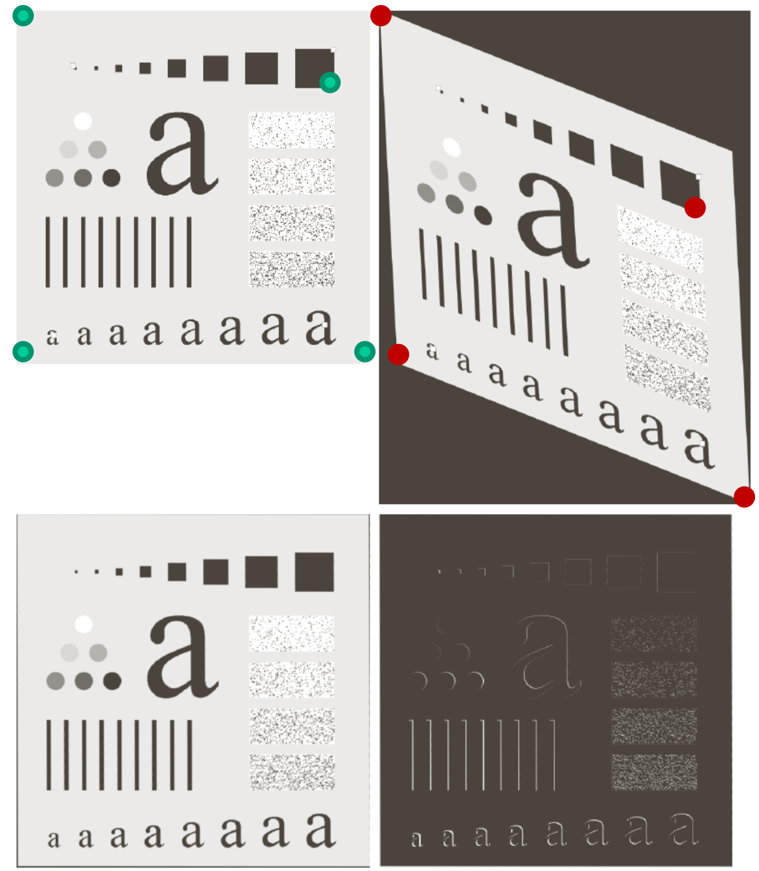
Image Registration

Compensate the geometric change in:

- view angle
- distance
- orientation
- sensor resolution
- object motion

Four major steps:

- Feature detection
- Feature matching
- Transformation model
- Resampling



a b
c d

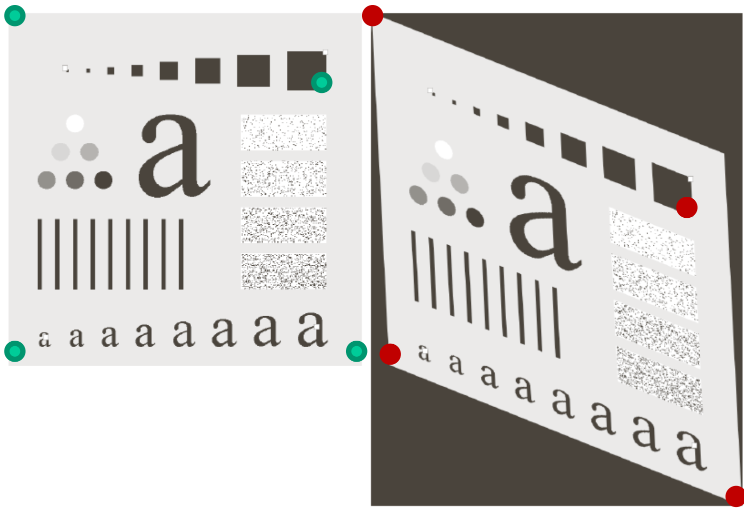
FIGURE 2.37 Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the borders). (d) Difference between (a) and (c), showing more registration errors.

Image Registration

Coordinates in the moving image (v, w)
Coordinates in the template image (x, y)

$$x = c_1 v + c_2 w + c_3 vw + c_4$$

$$y = c_5 v + c_6 w + c_7 vw + c_8$$



- **Known: coordinates of the points (x, y) and (v, w)**
- **Unknown: c_1 to c_8**

4 tie points -> 8 equations

Vector and Matrix Operations

$$\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

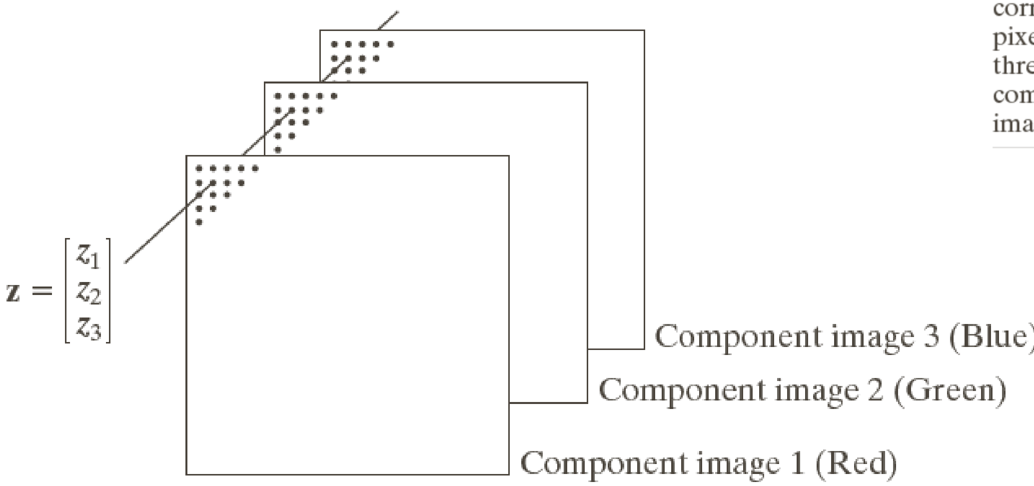


FIGURE 2.38 Formation of a vector from corresponding pixel values in three RGB component images.

$$D(\mathbf{z},\mathbf{a}) = \|\mathbf{z}-\mathbf{a}\| = \sqrt{(z_1-a_1)^2+(z_2-a_2)^2+\ldots+(z_n-a_n)^2}$$

Geometric transformations use vector and matrix operations

Spatial-Frequency Domain Transformation

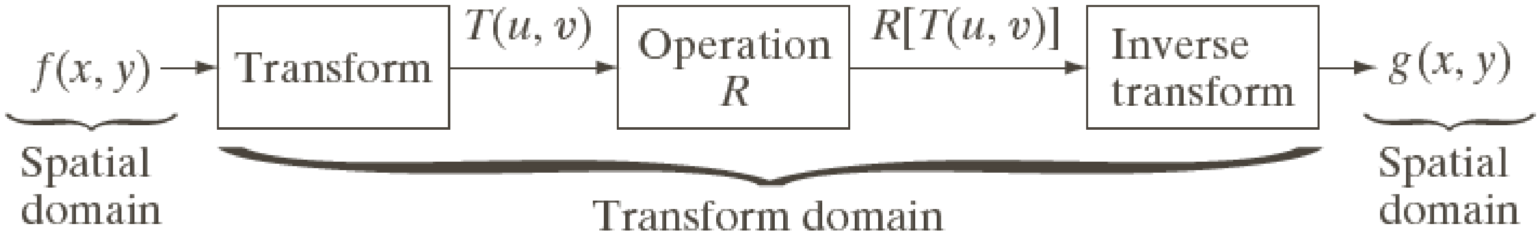


FIGURE 2.39 General approach for operating in the linear transform domain.

Forward transformation kernel

$$T(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) r(x,y,u,v),$$

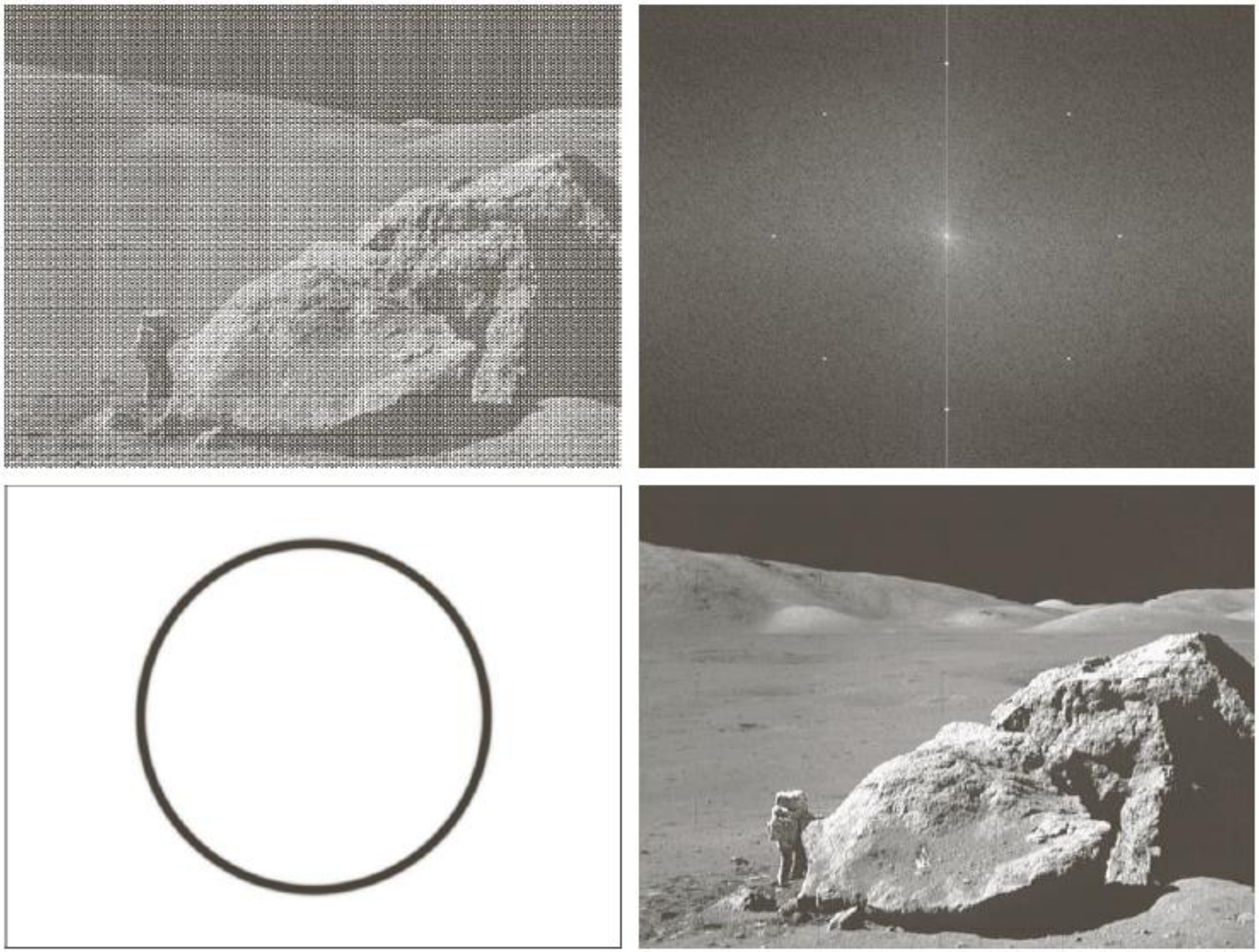
$$u = 0,1,\ldots, M-1$$
$$v = 0,1,\ldots, N-1$$

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u,v) s(x,y,u,v)$$

$$x = 0,1,\ldots, M-1$$
$$y = 0,1,\ldots, N-1$$

Inverse transformation kernel

Fourier Transforms and Filtering



a b
c d

FIGURE 2.40
(a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy of NASA.)

Fourier Transform

Separable kernel: $r(x, y, u, v) = r_1(x, u) r_2(y, v)$

Symmtric kernel: $r(x, y, u, v) = r_1(x, u) r_1(y, v)$

$$r(x, y, u, v) = e^{-j2\pi(ux/M + vy/N)}$$

$$s(x, y, u, v) = \frac{1}{MN} e^{j2\pi(ux/M + vy/N)}$$

Probability Methods

z_k is the kth intensity value
n_k is the number of pixels having the intensity value z_k

Probability of an intensity value

$$p(z_k) = \frac{n_k}{MN}, \quad \sum_{k=1}^{L-1} p(z_k) = 1$$

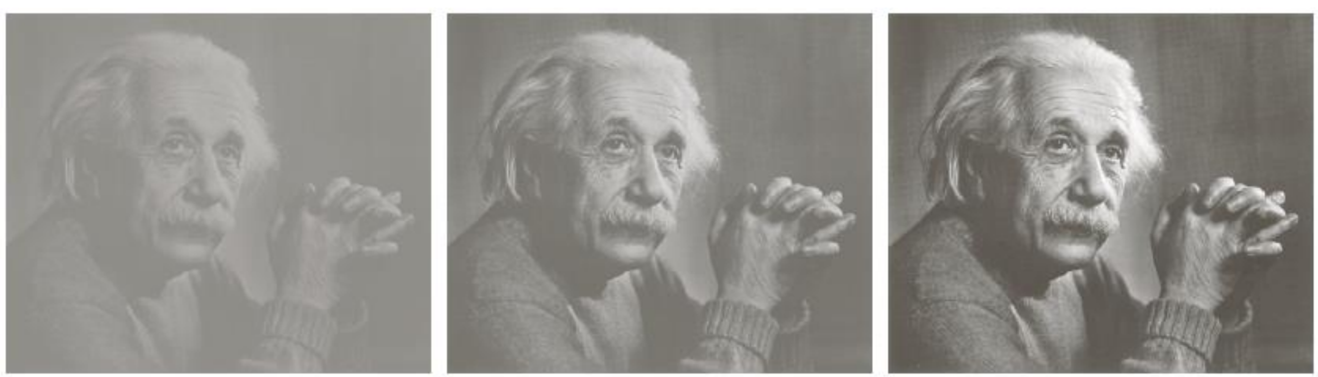
Probability Methods

$$m = \sum_{k=1}^{L-1} z_k \, p(z_k), \quad \sigma^2 = \sum_{k=1}^{L-1} (z_k - m)^2 \, p(z_k)$$

What do they mean?

$$\mu_n(z) = \sum_{k=1}^{L-1} (z_k - m)^n \, p(z_k)$$

n^{th} moment of z



a b c

FIGURE 2.41 Images exhibiting (a) low contrast, (b) medium contrast, and (c) high contrast.

Std=14.3

Std=31.6

Std=49.2

Stochastic Image-Sequence Processing

Using probability and random-process tools

Each pixel is a random event → each image frame is a random event, related to time

Probability plays a central role in modern image processing and computer vision

Summary

In this course, we will discuss all the concepts in details.

Now,

Intensity Transformation and Spatial Filtering

Reading: Chapter 3.

Spatial Domain

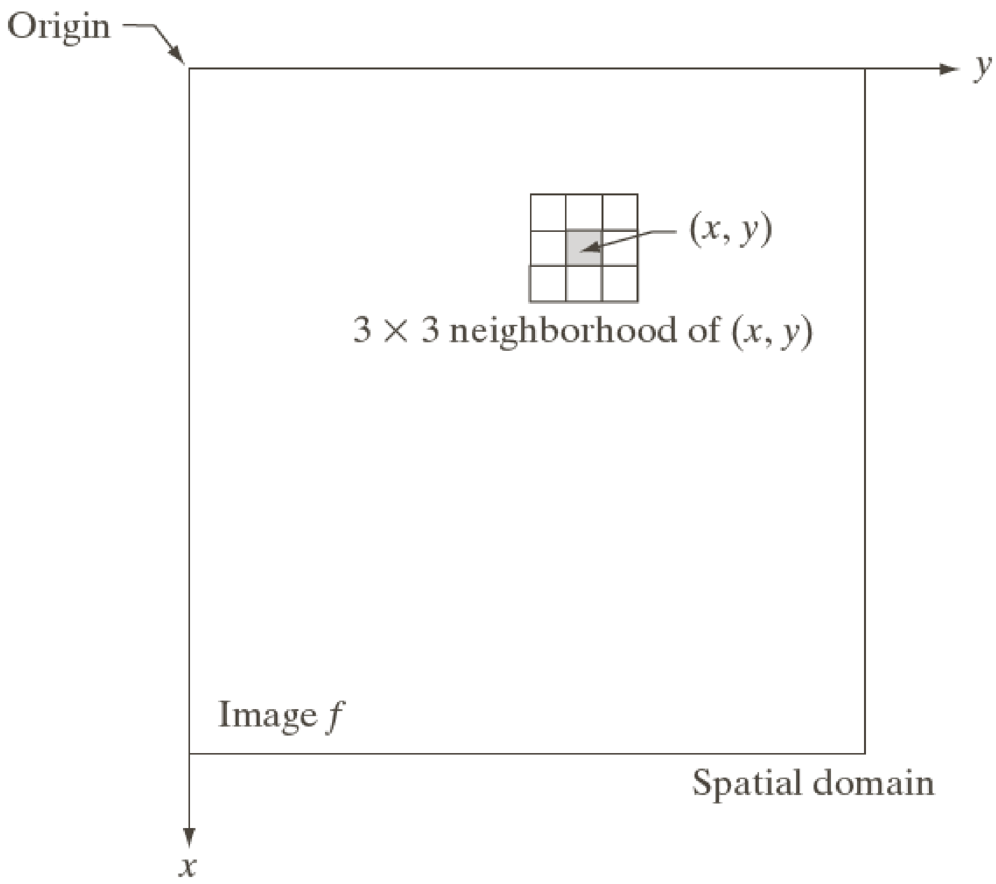


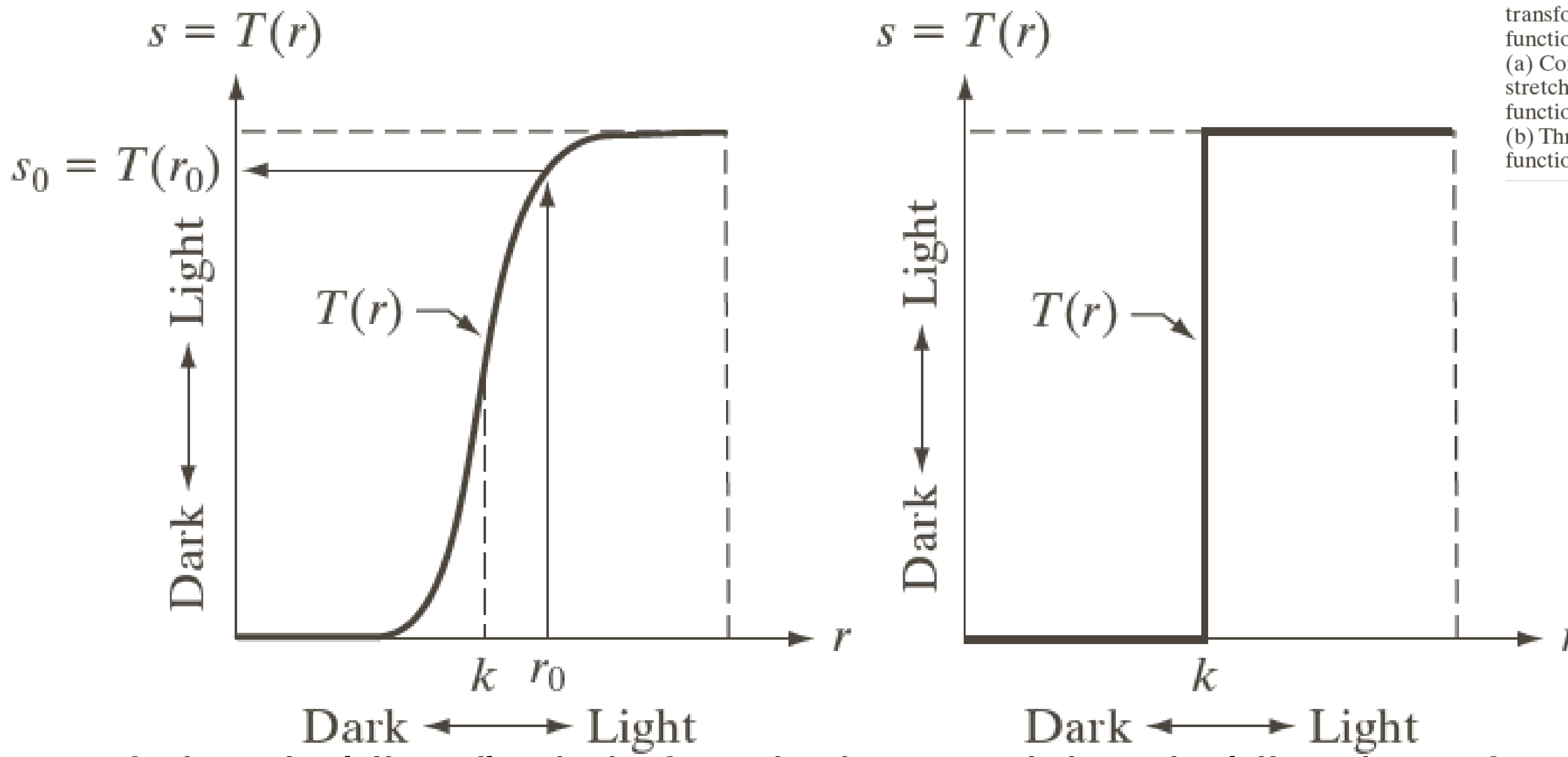
FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

$$g(x,y)=T\,[f(x,y)]$$

→ spatial filter

1x1 Neighborhood → Intensity Transformation → Image Enhancement

Contrast stretch



a b

FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

Soft thresholding (logistic function) Hard thresholding (step function)

$$\sim s = \frac{1}{1 + e^{r}}$$

Some Basic Intensity Transformation Functions

- **Thresholding – Logistic function**
- **Log transformation**
- **Power-law (Gamma correction)**
- **Piecewise-linear transformation**
- **Histogram processing**

Some Basic Intensity Transformation Functions

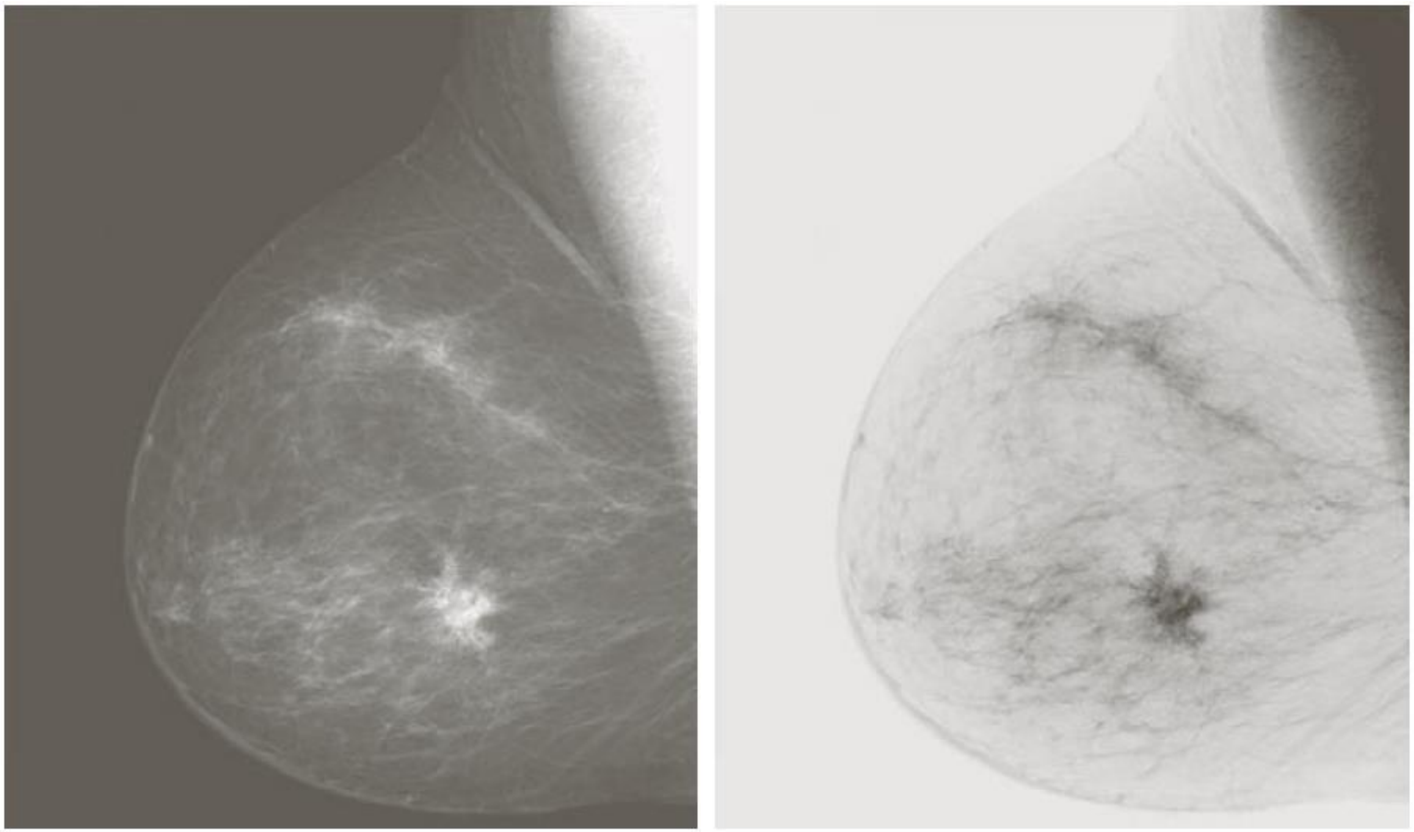


FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Image Negative: $s=L-1-r$

Basic Intensity Transformation Functions

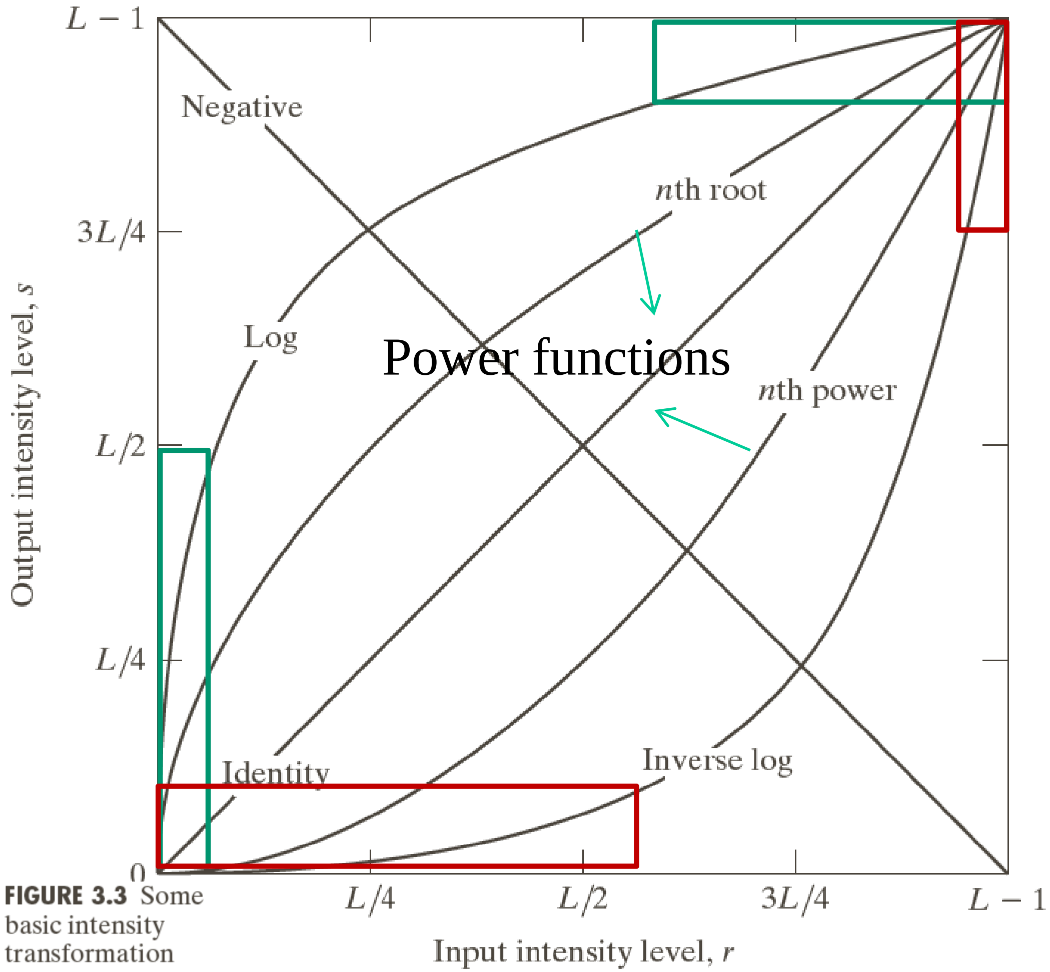


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Log function:

$$s = c\log(1 + r) \quad r \geq 0$$

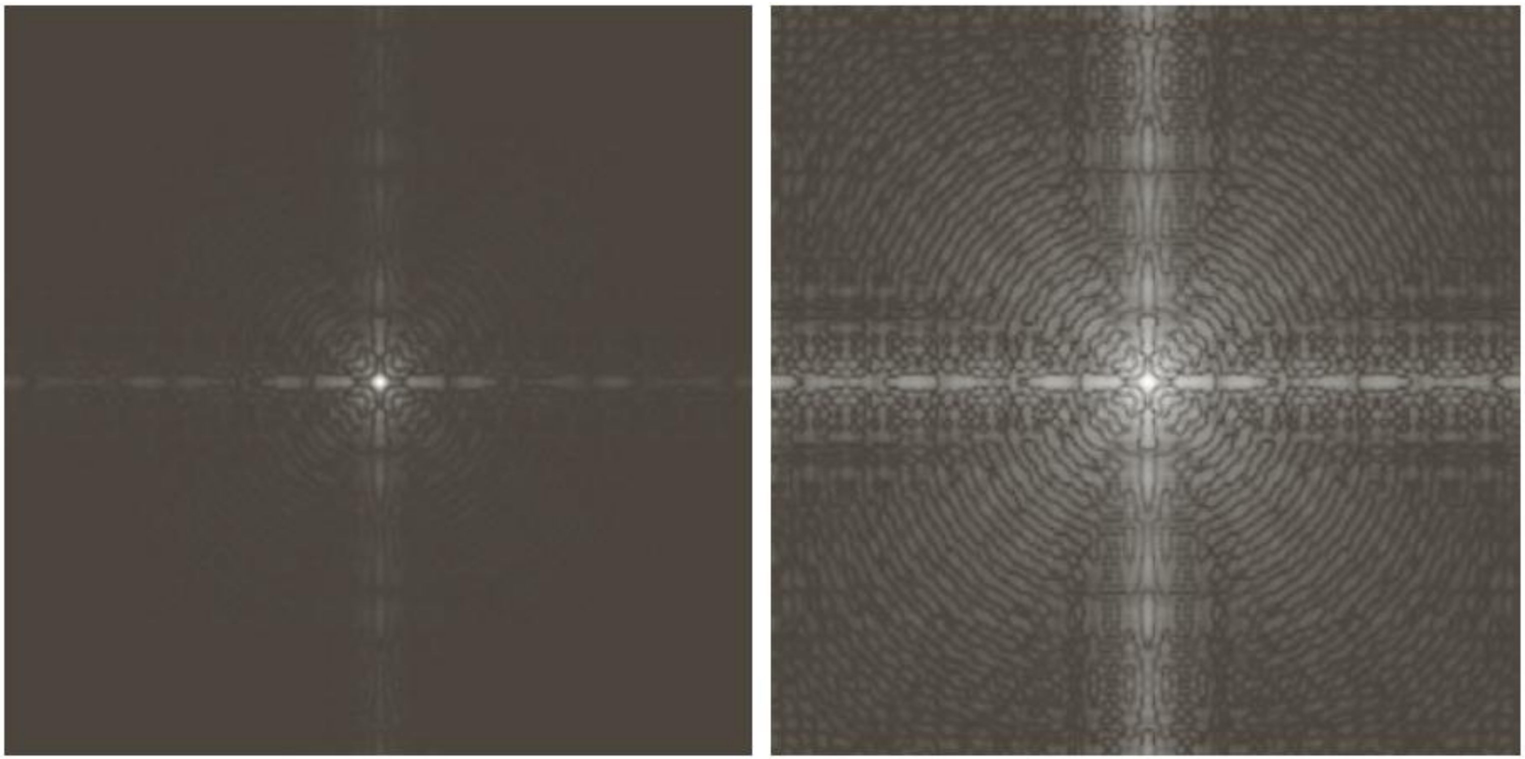
Stretch low intensity levels
Compress high intensity levels

Inverse log function:

$$s = c\log^{-1}(r)$$

Stretch high intensity levels
Compress low intensity levels

Log Transformations: $s = c \log(1+r)$



a b

FIGURE 3.5 (a) Fourier spectrum. (b) Result of applying the log transformation in Eq. (3.2-2) with $c = 1$.

Power-Law (Gamma) Transformations

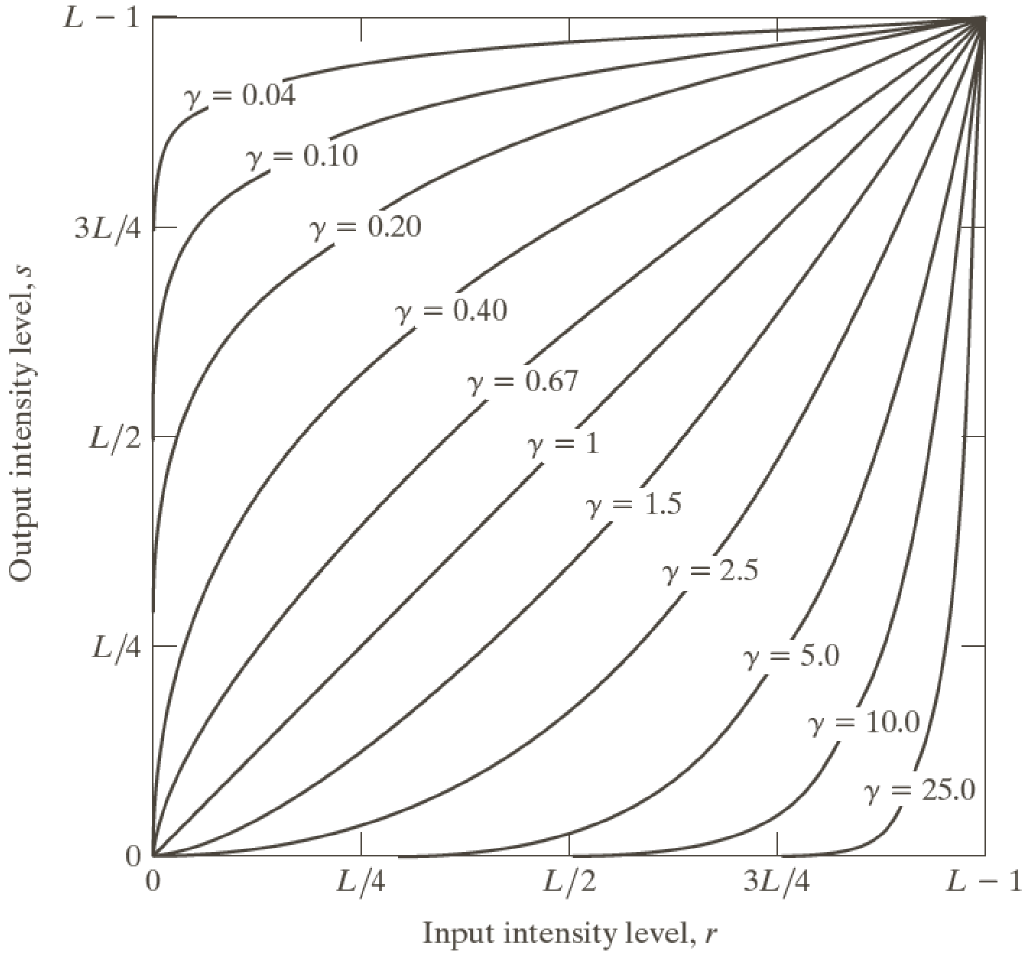
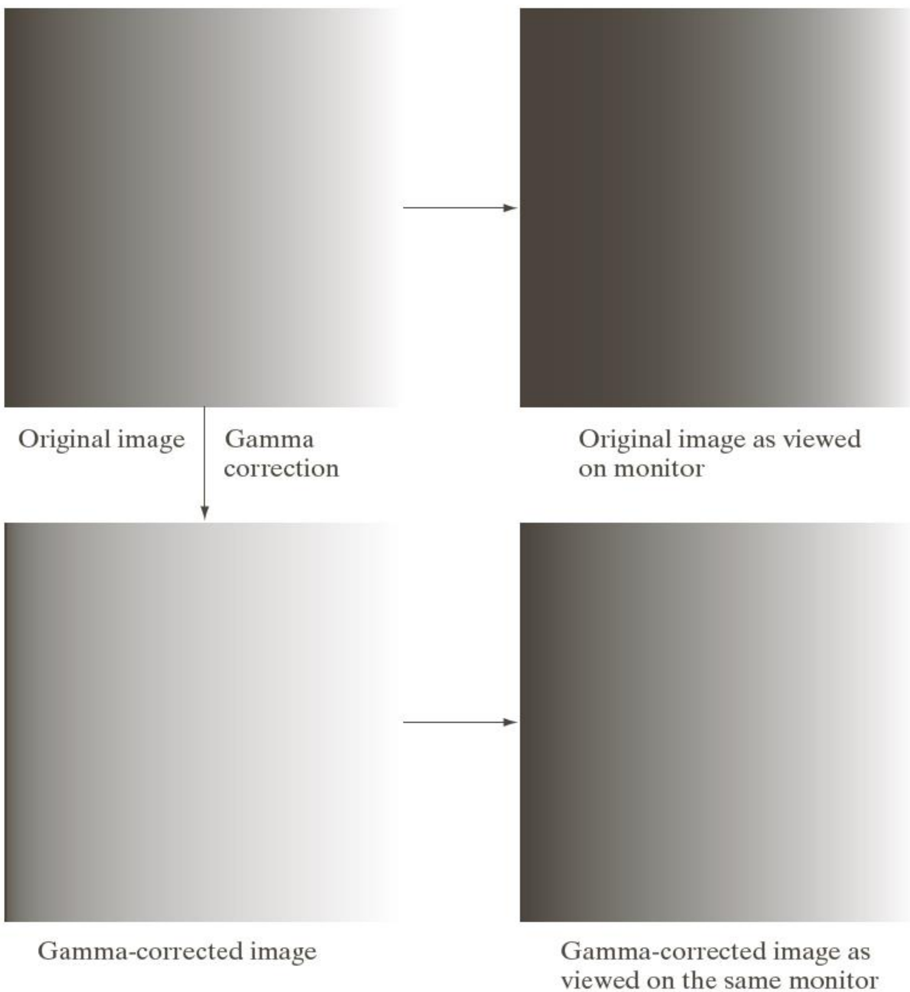


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

$$s = cr^{\gamma}$$

- More versatile than log transformation
- Performed by a lookup table

Power-Law (Gamma) Transformations



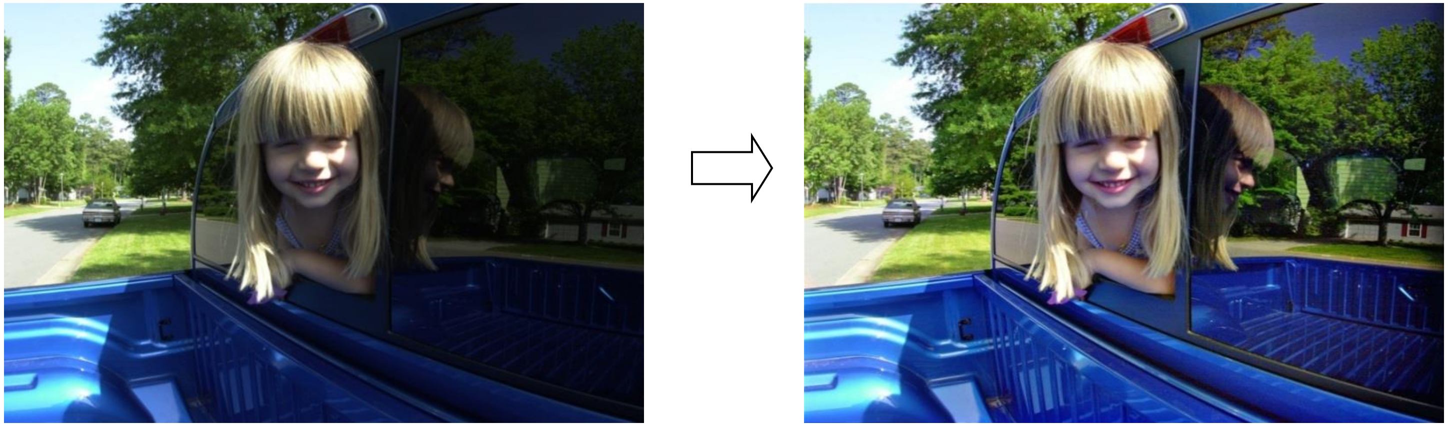
Monitors have an intensity-to-voltage response with a power function

$$s = r^{1/2.5}$$

a	b
c	d

FIGURE 3.7
(a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).

Image Enhancement Using Gamma Correction



Power-Law (Gamma) Transformations for Contrast Manipulation

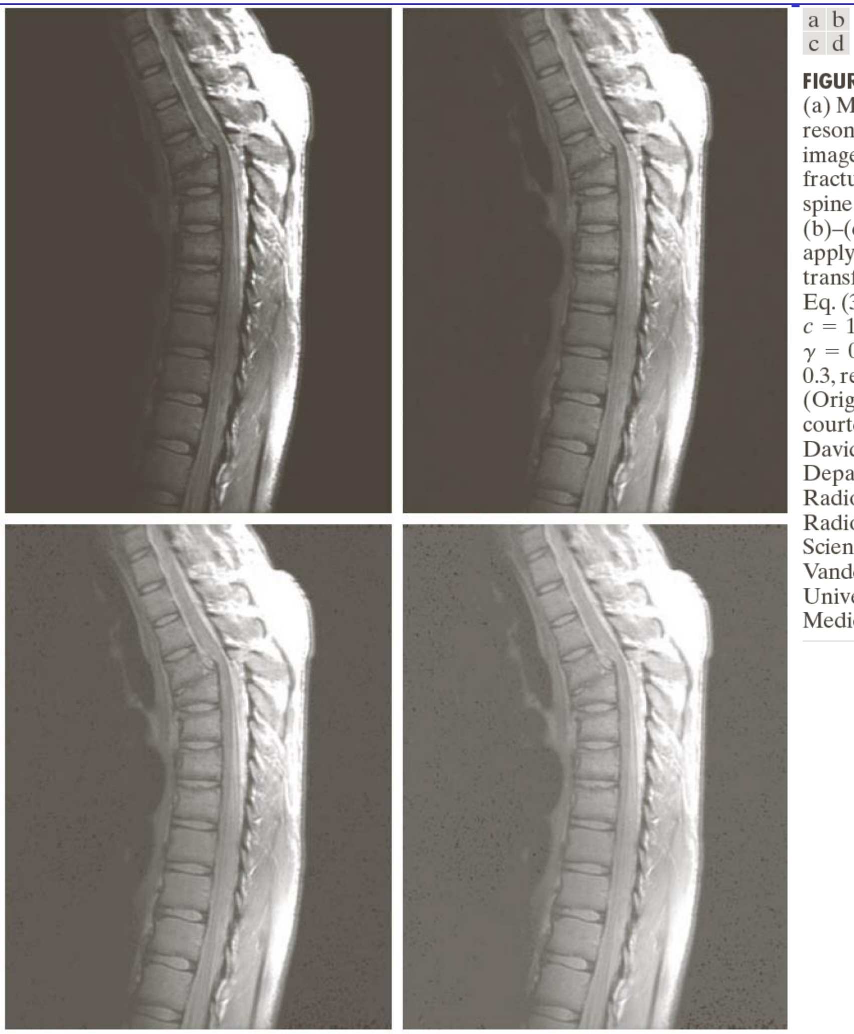


FIGURE 3.8 (a) Magnetic resonance image (MRI) of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Washed-out appearance caused by a small gamma value

Power-Law (Gamma) Transformations for Contrast Manipulation

a b
c d

FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

Washed-out appearance was reduced by a large gamma value